

I MCA B 2023
PYTHON
LAB EXERCISE – 1

1.

```
# Write a python program to count the frequency of any specific word (in your
# domain) in the paragraph.
paragraph="""I'm Pavitharani G P (2347244), has a zeal of innovation and
eager to continue exploring fields. In 2023, I ventured into the domain of
Photography prop store Management using Python. Its concept is about
acknowledging people about different categories of props available in town and
can order through this website. Passionate in the field of cyber security and
have keen interest in web designing"""

specificWord= input("Enter Word to find its frequency")
paragraph= paragraph.lower()
specificWord=specificWord.lower()
words = paragraph.split()
frequency = words.count(specificWord)
print("The word '",specificWord,"' Occurs ",frequency,"times")
```

OUTPUT :

The word ' in ' Occurs 4 times

2.

```
# # Write a python program to display all the datatypes of selected specific
# elements in the paragraph. (For example:- name - string, reg.no - int, marks
# float, etc.)
paragraph="""I'm Pavitharani G P (2347244), has a zeal of innovation and eager
to continue exploring fields. The project attributes various datatypes, IDs-
integer, names-string, description-string, Price-float. In 2023, I ventured
into the domain of Photography prop store Management using Python. Its concept
is about acknowledging people about different categories of props available in
town and can order through this website. Passionate in the field of cyber
security and have keen interest in web designing"""

import re

def datatypeExtractor(paragraph):
    dataSet={}
    regexPattern = r"([A-Za-z0-9_])\s*-\s*(integer|float|boolean|string)"
```

```

matches = re.findall(regexPattern, paragraph)

for elementName, dataType in matches:
    dataSet[elementName] = dataType
return dataSet

dataTypesSet = datatypeExtractor(paragraph)
for node,dtype in dataTypesSet.items():
    print(f"{node}-->{dtype}")

```

OUTPUT :

```

IDs-->integer
names-->string
description-->string
Price-->float

```

3.

```

# Write a python program to count the number of alphabets, numeric and other
# special symbols in the paragraph.
paragraph="""I'm Pavitharani G P (2347244), has a zeal of innovation and eager
to continue exploring fields. In 2023, I ventured into the domain of
Photography prop store Management using Python. Its concept is about
acknowledging people about different categories of props available in town and
can order through this website. Passionate in the field of cyber security and
have keen interest in web designing"""

alphabetCount=0
digitCount=0
specialCharCount=0
spaceCount=0
for charecter in paragraph:
    if(charecter.isdigit()):
        digitCount+=1
    elif(charecter.isalpha()):
        alphabetCount+=1
    elif(charecter.isspace()):
        spaceCount+=1
    else:
        specialCharCount+=1

print("Alphabets : ",alphabetCount)
print("Digits : ",digitCount)
print("Special Charecters : ",specialCharCount)
print("Spaces : ",spaceCount)

```

OUTPUT :

Alphabets : 314
Digits : 11
Special Charecters : 8
Spaces : 67

4.

```
# Create a Set with elements that consists of various data types (int, float,
# string, Boolean, etc. from your domain) and perform the functions pop(),
clear(),
# discard() and del. Write the insights as docstring.

def setOperation():
    initialSet={ 45 , True , 59.6 , "Sigma" , (9,5,6,7)}
    print("Initial Set is : ",initialSet)

    poppedElement= initialSet.pop()
    print("Popped Element : ",poppedElement)
    print("Set after Element Pop : ",initialSet)

    discardElement=45
    initialSet.discard(discardElement)
    print("Set after Discarding Element : ",initialSet)

    initialSet.clear()
    print("Set after Clearing : ",initialSet)

    initialSet={ 45 , True , 59.6 , "Sigma" , (9,5,6,7)}

    del initialSet

try:
    print("Set After deletion",initialSet)
except:
    print("Element Cant Be Displayed As it was deleted")
setOperation()
```

OUTPUT :

Element Cant Be Displayed As it was deleted
Initial Set is : {True, (9, 5, 6, 7), 'Sigma', 59.6, 45}
Popped Element : True
Set after Element Pop : {(9, 5, 6, 7), 'Sigma', 59.6, 45}
Set after Discarding Element : {(9, 5, 6, 7), 'Sigma', 59.6}
Set after Clearing : set()

5.

```
# Update the Set with minimum 5 string attributes of your domain and arrange
the
# Set in descending order.
"""The Attributes used here are :
CusName
CusId
Price
Rating
"""
domainAttributes={"CusName","CusID","Price","Rating"}
print("Initial Set : ",domainAttributes)
sortedSet= sorted(domainAttributes,reverse=True)
print("Sorted Set : ",sortedSet)
```

OUTPUT :

```
Initial Set :  {'Price', 'CusID', 'Rating', 'CusName'}
Sorted Set :  ['Rating', 'Price', 'CusName', 'CusID']
```

6.

```
# # Create a Tuple and Execute the packing and unpacking operations of tuples
using
# the attributes of your domain.
pattr1="CusId"
pattr2="CusName"
pattr3="Price"
pattr4="Product"
pattr5="Quality"
attributeTuple=tuple([pattr1,pattr2,pattr3,pattr4,pattr5])
attr1,attr2,*attr3,attr4=attributeTuple
print("The initial tuple : ",attributeTuple)
print("Attribute 1 : ",attr1)
print("Attribute 2 : ",attr2)
print("Attributes 3 : ",attr3)
print("Attribute 4 : ",attr4)
```

OUTPUT :

```
The initial tuple :  ('CusId', 'CusName', 'Price', 'Product',
'Quality')
Attribute 1 :  CusId
Attribute 2 :  CusName
Attributes 3 :  ['Price', 'Product']
Attribute 4 :  Quality
```

7.

```
# Enter your domain name as characters and count any number of characters and
print
# the count (for example - ('p','r','o','g','r','a','m') count of 'r' = 2

def alphabetCounter(domainName):
    letterSet={}
    for letter in domainName:
        if letter.isalpha():
            lowerLetter=letter.lower()
            letterSet[lowerLetter]=letterSet.get(lowerLetter,0)+1
    return letterSet

domainName=input("Enter the domain Name")
letterSet= alphabetCounter(domainName)

print(domainName)
print("Letter counts")
for letter,lCount in letterSet.items():
    print(letter," : ",lCount)
```

OUTPUT :

Photography Prop Store Management

Letter counts

p : 5
h : 2
o : 4
t : 3
g : 2
r : 3
a : 3
y : 2
s : 1
e : 4
m : 2
n : 2

8.

```
# # Enter your domain name, execute all the slicing possibilities and also
negative
# indexing.

domainName= input("Enter your Domain Name")
stringLength=len(domainName)
# print(stringLength)

for i in range(0,stringLength + 1):
    for j in range(i+1,stringLength + 1):
        subString = domainName[i:j]
        print(f"{domainName} [{i}:{j}] = {subString}")

for i in range(-stringLength-1,0):
    for j in range(i+1,0):
        subString = domainName[i:j]

print(f"{domainName} [{i}:{j}] = {subString}")
```

OUTPUT :

```
Photography Prop Store Management[0:1] = P
Photography Prop Store Management[0:2] = Ph
Photography Prop Store Management[0:3] = Pho
Photography Prop Store Management[0:4] = Phot
Photography Prop Store Management[0:5] = Photo
Photography Prop Store Management[0:6] = Photog
Photography Prop Store Management[0:7] = Photogr
Photography Prop Store Management[0:8] = Photogra
Photography Prop Store Management[0:9] = Photograp
Photography Prop Store Management[0:10] = Photograph
Photography Prop Store Management[0:11] = Photography
Photography Prop Store Management[0:12] = Photography
Photography Prop Store Management[0:13] = Photography P
Photography Prop Store Management[0:14] = Photography Pr
Photography Prop Store Management[0:15] = Photography Pro
Photography Prop Store Management[0:16] = Photography Prop
Photography Prop Store Management[0:17] = Photography Prop
Photography Prop Store Management[0:18] = Photography Prop S
Photography Prop Store Management[0:19] = Photography Prop St
Photography Prop Store Management[0:20] = Photography Prop Sto
Photography Prop Store Management[0:21] = Photography Prop Stor
Photography Prop Store Management[0:22] = Photography Prop Store
Photography Prop Store Management[0:23] = Photography Prop Store
Photography Prop Store Management[0:24] = Photography Prop Store M
Photography Prop Store Management[0:25] = Photography Prop Store Ma
```

...

Photography Prop Store Management[31:32] = n
Photography Prop Store Management[31:33] = nt
Photography Prop Store Management[32:33] = t
Photography Prop Store Management[-1:-1] = n